# Task: UCI The Great Escape



Stage II. Day 2. Source file uci.\* Available memory: 64 MB.

Al Bytone, the infamous thief, plans a bank robbery. He knows only too well that the moment he robs the bank a pursuit will be commenced. Unfortunately, Al Bytone is a poor driver and turning left causes him great trouble. This is why he tries to devise such an escape route that at each intersection he would either ride straight ahead or turn right. He is also aware that once he passes through any intersection, the police will come and remain there, waiting for him. Therefore he may pass through any intersection at most once. Furthermore, the police are always present at certain intersections, so Al Bytone will have to avoid these intersections as well (there's no police at the intersections near the bank and near Al Bytone's hideout.)

Al Bytone is planning his escape route. To your great (and rather unpleasant) surprise, he paid you a visit and told to calculate the number of different escape routes leading from the bank to his hideout complying the aforementioned requirements. Needless to say, Al Bytone does not take 'no' as an answer...

The streets of Byteburg form a rectangular grid. Every street runs either in the North-South or East-West direction, and every two non-parallel streets intersect. The bank is situated to the south of the south-western-most intersection. Al Bytone will start his great escape driving north.

#### Task

Write a programme that:

- reads from the standard input the location of hideout, descriptions of intersections with police and a positive integer *k*,
- calculates the number of different escape routes leading from the bank to the hideout complying the aforementioned requirements,
- writes out to the standard output this number's residue modulo *k*.

### Input

There are three integers in the first line of the standard input, n, m and k  $(1 \le n, m \le 100, 1 \le k \le 10^9)$ . The numbers n and m denote the number of streets leading in East-West and North-South direction, respectively. The second line contains two integers x and y  $(1 \le x \le m, 1 \le y \le n)$ . These represent the hideout's location — at the intersection of  $x^{th}$  street leading in North-South direction and  $y^{th}$  street leading in East-West direction. The streets are numbered from West to East and from North to South, from 1 to m and from 1 to n, respectively.

Each of the following *n* lines contains *m* characters ",\*" and/or ",+". This is the city map. The character in  $i^{th}$  line and  $j^{th}$  column of the map depicts the intersection of the  $i^{th}$  street leading from West to East with the  $j^{th}$  street leading from North to South. ",\*" means there is police at the intersection, while ",+" means there is no police there, so the escape route may pass through this intersection.

Al Bytone starts his great escape driving onto the intersection with coordinates (1,n) from the South, ie. from a nonexistent intersection (1, n + 1).

### Output

Your programme should write out the residue of the humber of escape routes modulo k in the first and former line of the standard output.

## Example

For the input data: 3 5 10 4 2 +++++ ++\*++